

Example

```

10 LPRINT CHR$(27); "2"; "* Normal spacing"
15 GOSUB 120
20 LPRINT CHR$(29); "F"; "* Auto spacing"
30 GOSUB 120
40 END
120 FOR I=20 TO 45 STEP 25
130 LPRINT CHR$(29); "H"; CHR$(0); CHR$(I); "Height="; I
140 FOR J=1 TO 2
150 LPRINT "Look at the line spacing!"
160 NEXT
170 LPRINT
175 NEXT
180 RETURN

```

```

* Normal spacing
Height= 20
Look at the line spacing!
Look at the line spacing!

```

```

Height= 45
Look at the line spacing!
Look at the line spacing!

```

```

* Auto spacing

```

```

Height= 20
Look at the line spacing!
Look at the line spacing!

```

```

Height= 45
Look at the line spacing!
Look at the line spacing!

```

ASCII code sequence

Decimal	(29) _D	(70) _D	(n)
Hexadecimal	<1B> _H	<46> _H	<n>

6.6 Commands Relating to Form Feeds**FF****(Form Feed)**

Name	Form Feed
-------------	-----------

Format	CHR\$(12);
---------------	------------

Function	To allow the paper to be changed.
-----------------	-----------------------------------

When the Form Feed code is executed, the paper is moved to the the change paper position, and the pens are capped. The ON-LINE LED flashes to show that the paper needs to be changed.

When the paper has been changed, and the ON-LINE button is pressed, the ON-LINE LED will cease to flash and the effective drawing area will be shown. Depending on whether the direction of printing is the portrait or landscape position, the pen will then be positioned at the head of the paper ready to print.

When a character is more than 2.4 mm high, it will be clipped if an attempt is made to print on the first line of a page.

Example	LPRINT CHR\$(12);
----------------	-------------------

ASCII code sequence

Decimal	(12) _D
Hexadecimal	<0C> _H

ESC C (n)

(Form length/lines)

Name

Number of lines per page.

Format

CHR\$(27) ; "C"; CHR\$(n);

Function

To set the number of lines of text on a page.

The number of lines is given by the value of (n).

The characters to be sent are the letter "C" (ASCII codes (67)_D, <43>_H) followed by the ASCII code which numerically equals the number of lines required.

The page length calculated as the current line spacing multiplied by the value of (n). This must not exceed the effective drawing area set by the plotter mode.

When a page is full, a form feed character is automatically sent.

This command is useful for setting the part of the page for printing text so that a further part can be used to plot diagrams, especially if the text is generated by a word processor.

Example

LPRINT CHR\$(27); "C"; CHR\$(30);

ASCII code sequence

Decimal	(27) _D (67) _D (n)
Hexadecimal	<1B> _H <43> _H <n>

Where n is the number of lines on the page.

ESC C (0) (n)

(Form length/inches)

Name

Set form length in inches.

Format

CHR\$(27) ; "C"; CHR\$(0); CHR\$(n);

Function

To set the page length in inches.

The length in inches is given by the value of (n).

The characters to be sent are the letter "C" (ASCII codes (67)_D, <43>_H) followed by ASCII code (0)_D, followed by the ASCII code which numerically equals the number of inches required.

The page length must not exceed the effective drawing area set by the plotter mode. Since the effective drawing area is set in 0.1mm, the value of $n \times 254$ must be less than the value used to set the drawing area. care should be taken that the correct value is taken if the printing is carried out in the landscape position.

When a page is full, a form feed character is automatically sent.

This command is useful for setting the part of the page for printing text so that a further part can be used to plot diagrams, especially if the text is generated by a word processor. The form feed will be sent when the page length is reached, and the page length need not be the full length of the paper.

Example

LPRINT CHR\$(27); "C"; CHR\$(0); CHR\$(11);

ASCII code sequence

Decimal	(27) _D (67) _D (0) _D (n)
Hexadecimal	<1B> _H <43> _H <0> _H <n>

Where n is the length of the page in inches.

ESC N (n)

(Skip-over-perforation)

Name Set skip-over-perforation.

Format CHR\$(27) ; "N"; CHR\$(n);

Function To set the number of lines at the base of the page which are to be skipped.

This command is added to allow consistency with other EPSON printers. In most cases fanfold paper is used on such a printer. When a continuous piece of text is sent to the printer under the control of an applications program such as a word processor, the word processor can control the number of lines of text and send a form feed when the bottom of the page is reached. If the number of lines set by the word processor software is less than the number of lines on the page, a number of blank lines will be left at the base of the page. When text such as a program listing is sent, the text just keeps printing and could continue over the perforation, making it difficult to read. This command makes it possible to send a form feed when the text is within a specified number of lines of the bottom of the page. Although it is possible to use perforated continuous paper on the HI-80, the paper is treated as if it were single sheet. The paper has to be changed manually, moved up and the ON-LINE button pressed to continue onto the next sheet. There is thus no possibility of writing continuously and so covering the perforation. It is often convenient to have some blank lines at the base of the page, and so this command should be seen as a command to set these lines.

When the text reaches the specified number of lines from the base of the form, a form feed character is automatically sent.

ASCII code sequence

Decimal (27)_D (78)_D (n)
Hexadecimal <1B>_H <4E>_H <n>

Where n is the number of lines to skip at the base of the page.

ESC O

(Cancel Skip-over-perforation)

Name Cancel skip-over-perforation

Format CHR\$(27);"O";

Function To reset the bottom of form set by the <ESC> N (n) sequence.

When this command is executed, the bottom of the page is set to the limit of the effective drawing area, rather than the number of lines from the bottom of the form set by the <ESC> N (n) command sequence.

ASCII code sequence

Decimal (27)_D (79)_D
Hexadecimal <1B>_H <4F>_H

6.7 Commands Relating to Formatting Text

BS

(Back Space)

Name Back space.

Format CHR\$(8);

Function To move the character printing position back by one character.
The amount which the pen moves is equal to the width of one character.
This allows characters to be printed on top of one another or for the pen to be positioned at a particular place on a line after characters have already been printed.

If a BS is sent to the HI-80 and the pen is at the beginning of the line, nothing will happen.

Example

```

10 GOSUB 100
20 LPRINT "====="
30 LPRINT CHR$(14);
40 GOSUB 100
45 LPRINT CHR$(20);
50 LPRINT "====="
60 LIST
70 END
100 LPRINT "YYYYY";
110 LPRINT CHR$(8);CHR$(8);
120 RETURN

YYYYY=====
Y Y Y Y Y=====
    
```

ASCII code sequence

Decimal	(08) _D
Hexadecimal	<08> _H

DEL

(Delete)

Name Delete

Format CHR\$(127);

Function To delete the last character in the buffer.

The ASCII code set contains a character which effectively has the same function as the backspace, but with the code (127)_D or <7F>_H. Some computers have a key which produces this ASCII code. Pressing a DEL key on the keyboard may delete the character under the cursor rather than the one to the left of the cursor. However, when functioning as a printer command, it always deletes the previous character in the buffer.

Example LPRINT "DELETE"; CHR\$(127); "ING"

would print:

DELETING

ASCII code sequence

Decimal	(127) _D
Hexadecimal	<7F> _H

ESC I (n)

(Left Margin)

Name Left Margin

Format CHR\$(27) ; "I" ; CHR\$(n);

NOTE:

This command uses a lower case "i" as in "Hello".

Function

To set the margin on the left hand side of the paper so that printing begins at a specified number of character positions. The number of character positions is specified by the value of (n) and denotes the number of characters of the current character width.

If the value of (n) given is greater than the setting of the right margin, the command is ignored.

When a carriage return is executed, the pen will move to the position of the left margin.

If the pen is not at the margin, it will be moved to the new setting of the margin.

The left margin is set at the edge of the effective drawing area on power up.

Example LPRINT CHR\$(27); "I"; CHR\$(5);

ASCII code sequence

Decimal	(27) _D (108) _D (n)
Hexadecimal	<1B> _H <6C> _H <n>

Where (n) is the number of characters to set the left margin.

ESC Q (n)

(Right Margin)

Name Right Margin

Format CHR\$(27) ; "Q" ; CHR\$(n);

Function

To set the margin on the right hand side of the paper so that printing ends at a specified number of character positions from the right. The number of character positions is specified by the value of (n) and denotes the number of characters of the current character width. The position of the margin is measured from the left hand side of the paper.

If the value of (n) given is less than the setting of the left margin, the command is ignored.

If the characters are enlarged to such an extent as to be greater than the size of the interval between the left and right hand margins, the right hand margin setting will be changed.

The right margin is set at the edge of the effective drawing area on power up. When data would be printed over the edge of the margin a carriage return/line feed is automatically given.

Example LPRINT CHR\$(27); "Q"; CHR\$(45);

ASCII code sequence

Decimal	(27) _D (81) _D (n)
Hexadecimal	<1B> _H <51> _H <n>

Where (n) is the number of characters to set the right margin.

ESC e (0) (n)

(HTAB setting)

Name Horizontal tab unit setting

Format CHR\$(27) ; "e"; CHR\$(0); CHR\$(n);

Function To set the relative horizontal tab positions.

The tab positions are set at every (n) character positions across the page.

To cause text to be printed at a particular tab position, the HT character (ASCII code (9)_D) must be given before the text characters to be printed at a particular position.

If the left margin is changed, the position of the tab settings will be relative to the new margin.

Example

```
5 LPRINT "12345678901234567890123456789"
10 LPRINT CHR$(27); "e"; CHR$(0); CHR$(6);
20 FOR I=1 TO 3
30 LPRINT CHR$(9); "TAB";
40 NEXT
```

```
12345678901234567890123456789
   TAB   TAB   TAB
```

ASCII code sequence

Decimal	(27) _D (101) _D (0) _D (n)
Hexadecimal	<1B> _H <65> _H <0> _H <n>

where (n) is the relative tab unit setting.

HT

(Horizontal Tab)

Name Horizontal tabulation.

Format CHR\$(9);

Function To cause the carriage to skip to the horizontal tab position set by the <ESC> e (0) (n) command sequence.

ASCII code sequence

Decimal	(9) _D
Hexadecimal	<9> _H

ESC f (0) (n)

(Skip Horizontal)

Name Skip to horizontal tab position

Format CHR\$(27); "f"; CHR\$(0); CHR\$(n);

Function To cause the carriage to skip to a specified horizontal position.

The <ESC> e (0) (n) command allows the horizontal tabs to be set at a fixed interval across the page, and then the HT character (ASCII code (9)_D) skips to that position before printing the next string of text. This command sequence allows the pen to be moved to a specific horizontal position in one particular case. The position is given by the value of (n) and is counted from the left margin, in character columns according to the current character width.

ASCII code sequence

Decimal	(27) _D (102) _D (00) _D (n)
Hexadecimal	<1B> _H <66> _H <00> _H <n>

where (n) is the horizontal position to skip to.

ESC e (1) (n)

(VTAB setting)

Name Vertical tab unit setting

Format CHR\$(27); "e"; CHR\$(1); CHR\$(n);

Function To set the relative vertical tab positions.

The tab positions are set at every (n) character positions down the page.

To cause text to be printed at a particular tab position, the VT character (ASCII code (11)_D) must be given before the text characters to be printed on a particular line.

If the line spacing is changed, the line spacing for tab settings will remain at the old line spacing. Thus if it is necessary to tab vertically with same relative number of lines at the new line spacing, the command must be given again.

If the value of (n) is greater than the number of lines on the page, the command will be ignored.

If the height of the characters is increased, the line spacing will be based on the new character height.

Example

```
10 LPRINT CHR$(27); "e"; CHR$(1); CHR$(4);
20 FOR I=0 TO 3
30 LPRINT CHR$(11); "Vertical TAB";
40 NEXT
```

Vertical TAB _____ 4th line

Vertical TAB _____ 8th line

Vertical TAB _____ 12th line

ASCII code sequence

Decimal
Hexadecimal

(27)_D (101)_D (1) (n)
<27>_H <65>_H <1>_H <n>

where (n) is the relative tab unit setting.

VT

(Vertical Tab)

Name

Vertical tabulation.

Format

CHR\$(11);

Function

To cause the carriage to skip to the vertical tab position set by the <ESC> e (1) (n) command sequence.

When the VT code is executed, the pen is moved to the left margin to print the new line.

NOTE:

This command will also cancel enlarged mode characters set by the SO code (ASCII code 14) or the <ESC> SO sequence since it is effectively executing multiple line feeds.

ASCII code sequence

Decimal

(11)_D

Hexadecimal

<0B>_H

ESC f (1) (n)

(Skip Vertical)

Name

Skip to vertical tab position

Format

CHR\$(27); "f"; CHR\$(1); CHR\$(n);

Function

To cause the advance the paper to print on a specified line.

The <ESC> e (1) (n) command allows the vertical tabs to be set at fixed intervals down the page, and then the VT character (ASCII code (11)_D) skips to that line before printing the next string of text. The <ESC> f (1) (n) command sequence allows the pen to be moved to a specific line at one particular time. The line number is given by the value of (n).

ASCII code sequence

Decimal

(27)_D (102)_D (1)_D (n)

Hexadecimal

<1B>_H <66>_H <01>_H <n>

where (n) is the vertical line to skip to.

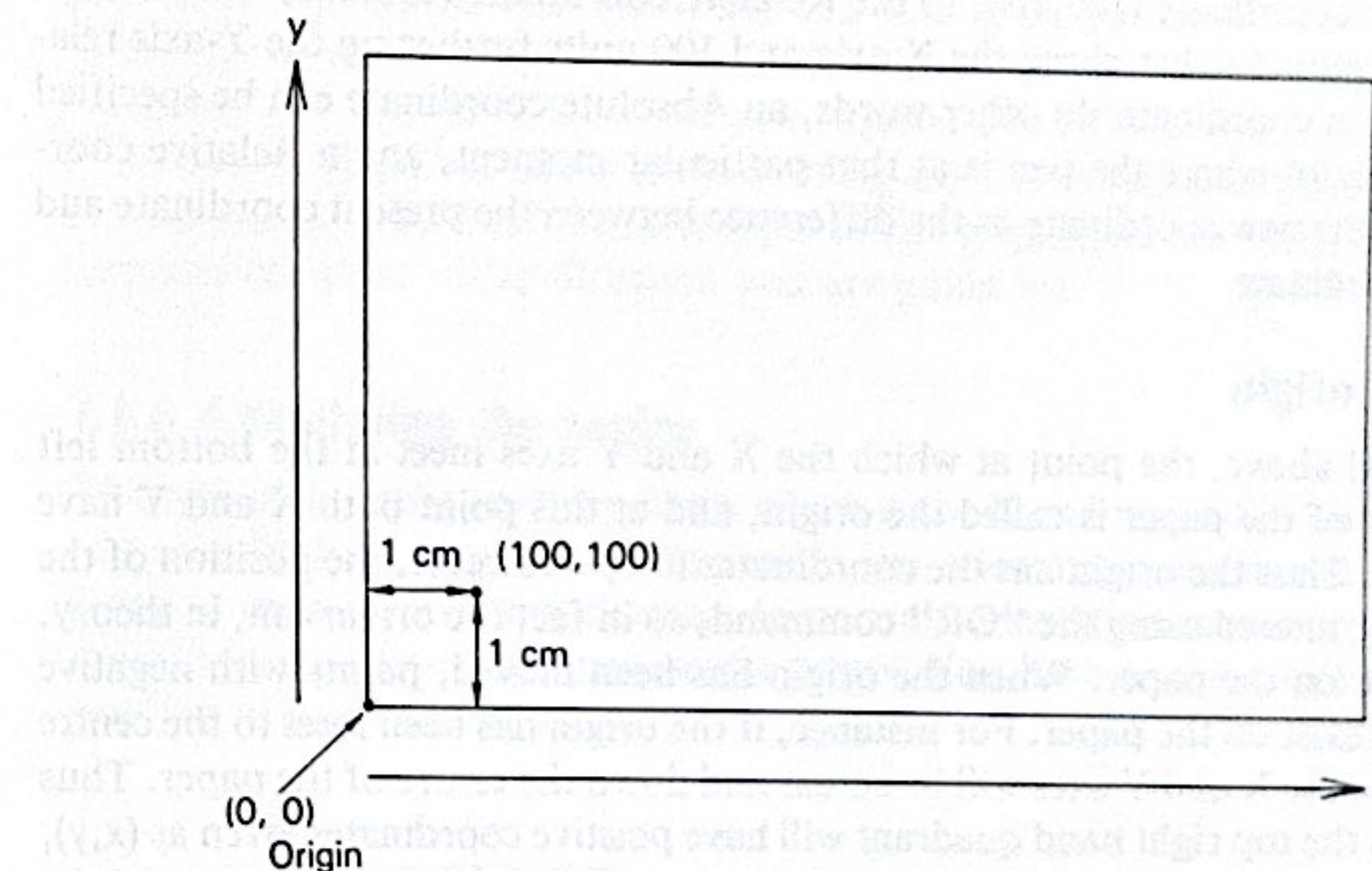
Chapter 7 GENERAL PLOTTING CONCEPTS AND SOME EXAMPLE PROGRAMS

This chapter is intended to show you some of the ways that the HI-80 can be used and to explain the mathematics behind plotting in simple terms.

7.1 Plotting Concepts

7.1.1 X and Y axes

In order to locate the position of the pen the HI-80 plotter uses a system called Cartesian coordinates. This means that a particular point on the paper can be defined by giving its distance from each of two axes at right-angles to each other. These axes are called the X-axis and Y-axis. To illustrate this, take a piece of ordinary A4 paper and lay it in front of you with the long sides at top and bottom and the short sides at left and right. The bottom left hand corner is called the 'origin', the left hand short edge is the Y-axis and the bottom long edge nearest you is the X-axis. Any particular point on the paper can be defined by giving its position as a distance along the X-axis followed by a distance along the Y-axis. Thus a point given the coordinates (5,7) will be in a position that corresponds to 5 units along the X-axis and 7 units up the Y-axis. The units used on the HI-80 are 0.1mm, so a point located 1cm along from the left hand edge and 1cm up from the bottom edge can be described as having the coordinates (100,100), that is, an X-coordinate of $100 \times 0.1\text{mm}$ and a Y-coordinate of $100 \times 0.1\text{mm}$.



These axes are always in the same direction as far as the HI-80 is concerned. When you insert paper into the plotter you will find that the paper is moved in the direction of the X-axis and the pen holder moves in the direction of the Y-axis, which makes the relationship easier to remember.

7.1.2 Designing a layout

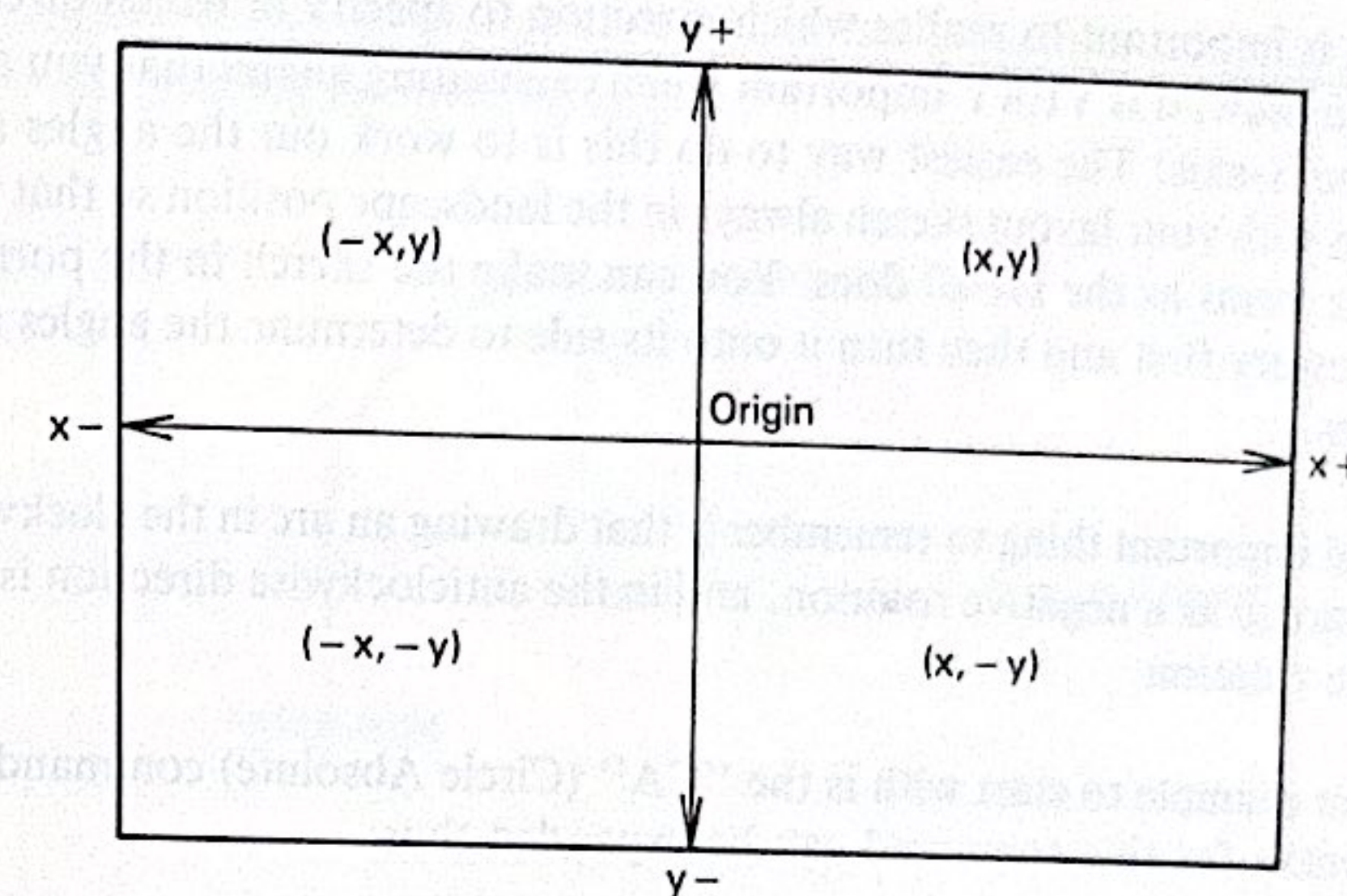
A considerable amount of the work you may do on the plotter will require some preparation in designing the layout you want before starting the actual programming. It is a good idea to obtain some squared paper or graph paper ruled in millimetres or multiples of millimetres on which to draft your designs. The HI-80 draws in increments of 0.1mm so it is possible to obtain some quite intricate designs. Obviously, drafting in steps of 0.1mm is very difficult, so you can enlarge your design by a factor of 10 in order to see the fine detail, then adjust the values you give in the plotter commands in order to reduce the design to its correct size. Using this method it is possible to draw complicated maps and intricate designs with the plotter pen movements giving lines only 0.1mm long, although your original design had lines 1mm long. When using fine-tipped pens the output will look superb and every bit as good as output from a much more sophisticated and expensive machine.

7.1.3 Relative and absolute coordinates

Two terms commonly used in plotter commands are RELATIVE and ABSOLUTE. These sometimes cause a certain amount of confusion. An Absolute coordinate is one which is given in relation to the origin (which is usually the bottom left hand corner of the paper). A Relative coordinate is one which is given as the difference between one location and another. For instance, a move from the Absolute coordinate (100,100) to the Absolute coordinate (200,200) is the same as a move from the Absolute coordinate (100,100) to the Relative coordinate (100,100). That is, the move is 100 units further along the X-axis and 100 units further up the Y-axis relative to the first coordinate. In other words, an Absolute coordinate can be specified independently of where the pen is at that particular moment, and a Relative coordinate gives the new coordinate as the difference between the present coordinate and the new coordinate.

7.1.4 The origin

As explained above, the point at which the X and Y axes meet at the bottom left hand corner of the paper is called the origin, and at this point both X and Y have a value of 0. Thus the origin has the coordinates (0,0). However, the position of the origin can be altered using the "OR" command, so in fact the origin can, in theory, be anywhere on the paper. When the origin has been moved, points with negative coordinates exist on the paper. For instance, if the origin has been reset to the centre of the paper, the X and Y axes will be across and down the centre of the paper. Thus any point in the top right hand quadrant will have positive coordinates given as (x,y), in the top left hand quadrant the X coordinate is negative giving coordinates of the form (-x,y). In the bottom left hand quadrant both coordinates are negative (-x,-y) and in the bottom right quadrant the Y coordinate is always negative giving coordinates of the type (x,-y).



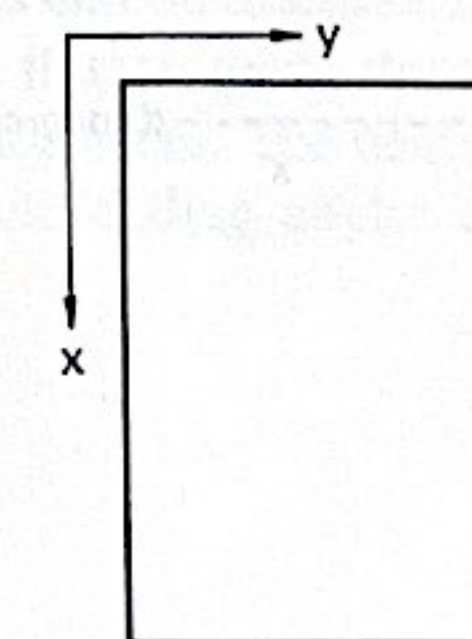
7.1.5 Working out the angles for arcs, circles, pie segments and labels

If you are not a mathematician, you may find the following explanation useful when working out the starting and ending angles of your circles and arcs.

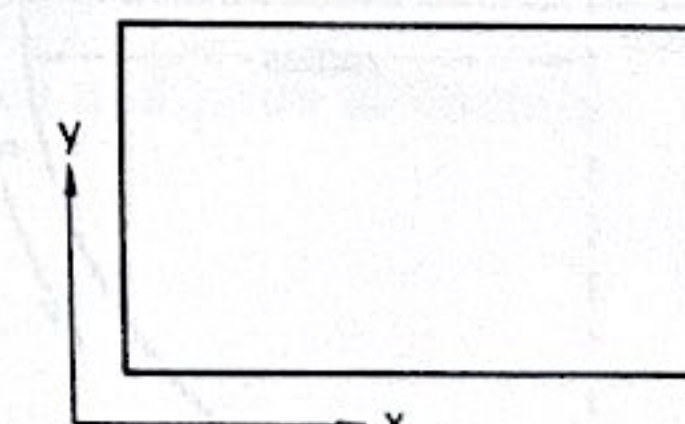
First of all, remember that the string and ending angles are absolute, not relative, even when using the "CR" command. In other words, having worked out the starting angles and the arc of the circle you require, DO NOT give the amount of curve as the ending angle but add it to the starting angle to obtain the ending angle. When you do this be very careful of the signs of the angles you are specifying and make sure you are clear which direction you are going in.

7.1.6 Calculating the angles

To begin with, make sure you know which is the X and which the Y-axis, otherwise you may find the arc does not appear in the orientation you intended. If you are going to draw a picture or diagram in portrait orientation remember that the axes are still in the same directions on the paper. You have to think in terms of which way the picture is to be drawn.



Portrait position



Landscape position

Whereas it is important to realise which direction to specify in which direction you are going to draw, it is VERY important when calculating angles that you are SURE which is the X-axis! The easiest way to do this is to work out the angles and point coordinates with your layout sketch always in the landscape position so that you think in the same terms as the HI-80 does. You can make the sketch in the portrait position if necessary first and then turn it onto its side to determine the angles and point coordinates.

The second important thing to remember is that drawing an arc in the clockwise direction is regarded as a negative rotation, and in the anticlockwise direction is regarded as positive rotation.

The easiest example to start with is the "CA" (Circle Absolute) command in Mode 0. The syntax for this command can be expanded thus:

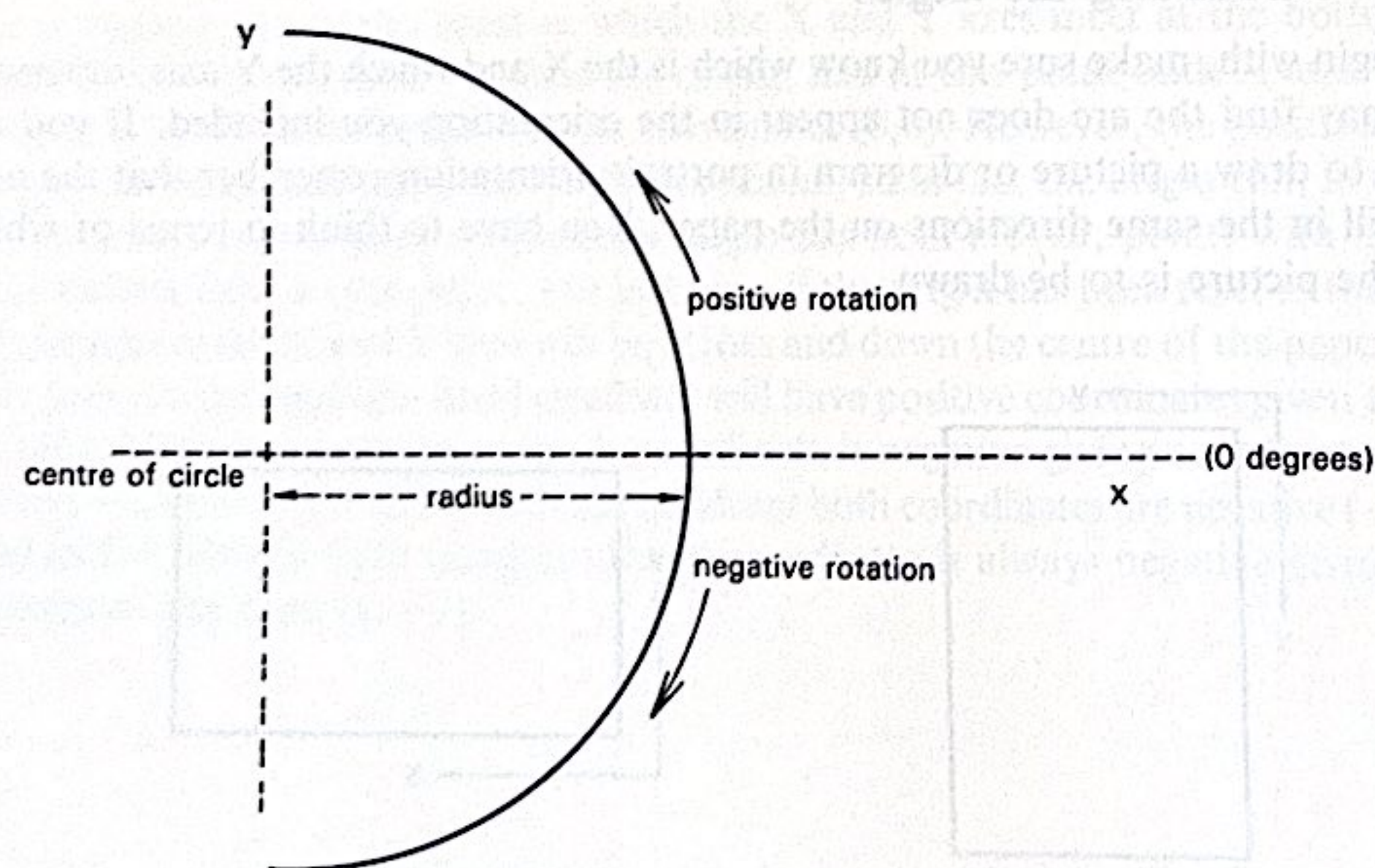
"CA coordinates of centre of circle, radius, starting angle, ending angle"

The command summary shows it as:

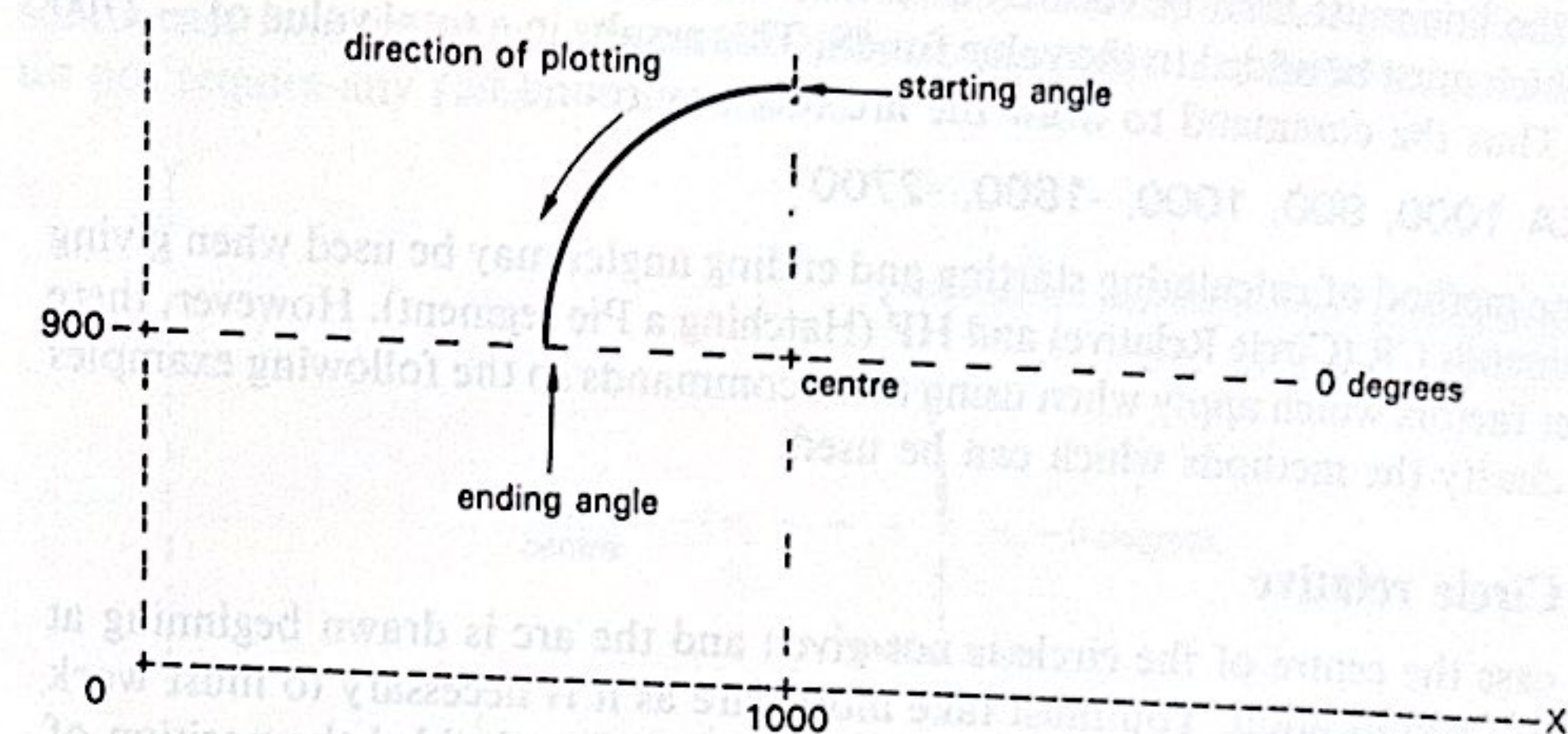
"CAx,y,r,θS,θE"

The coordinates of the centre of the circle are given in the normal way and are absolute values, that is, calculated from the origin and not from the current pen location. The radius is given in units of 0.1mm, so for a 1cm radius the value will be given as 100.

The starting and ending angles are given in units of 0.1 degrees, so a 90 degree angle will be given as 900. These angles are calculated relative to a line passing through the centre of the circle, parallel to the X-axis. This is regarded as 0 degrees. Positive angles are those which lie on the arc generated when this line is rotated anticlockwise, and negative angles are those generated when the line is rotated clockwise:



The next diagram shows an arc and is followed by a detailed explanation of how to calculate the starting and ending angles.



The values to be calculated are, in order:

x,y (coordinates of the centre of the circle). This is easily worked out from the diagram - (1000,900).

r (radius of the circle in 0.1mm). For instance we can use a value of 1000 to give an actual radius of 10cm.

θS (starting angle in 0.1 degree). Remember that 0 degrees is along the line parallel to the X-axis and passing through the centre of the circle. In order to reach the starting angle the radius line has to be rotated through 90 degrees in an anticlockwise direction. This gives a value for θS of 900.

θE (ending angle). To reach this angle the radius line must be rotated through a further 90 degrees anticlockwise. This value must be added to the starting angle value, giving a final value of 1800 for the ending angle.

Thus the complete command to draw this arc is:

"CA 1000, 900, 1000, 900, 1800"

The angles can be calculated in a different way to draw the same arc in the opposite direction, that is, using the ending angle of the previous example as the starting angle and vice versa. The centre coordinates and radius will remain the same, but the starting and ending angles will be calculated and given differently.

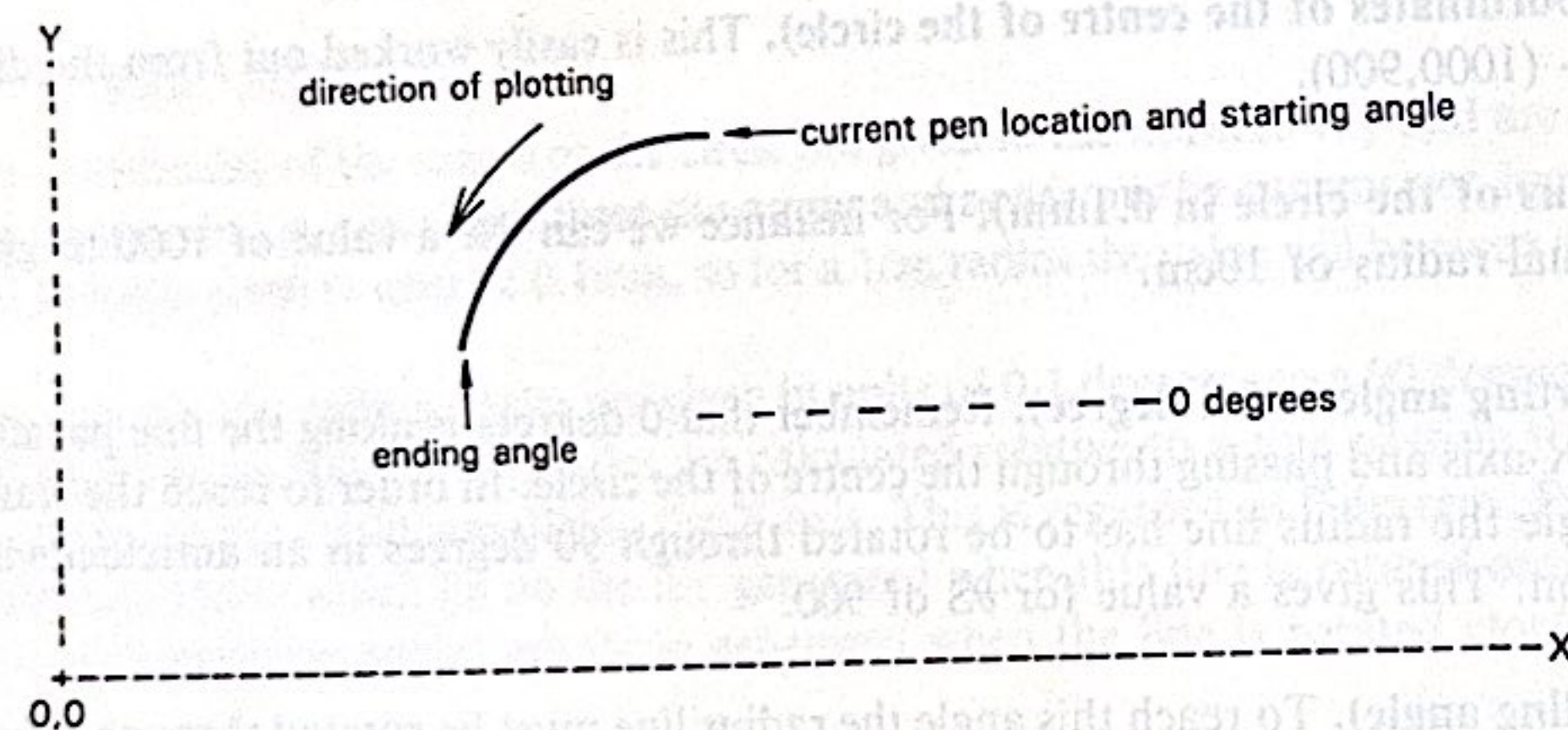
This time the radius line at 0 degrees must be rotated clockwise (the negative direction) 180 degrees to reach the starting angle, giving a value for θS of -1800 . To obtain θE the line must then be rotated a further 90 degrees clockwise, and value of -900 which must be added to the value for θS . This results in a total value of -2700 for θE . Thus the command to draw the arc this way round is:

`"CA 1000, 900, 1000, -1800, -2700"`

The same method of calculating starting and ending angles may be used when giving the commands CR (Circle Relative) and HP (Hatching a Pie segment). However, there are other factors which apply when using these commands so the following examples should clarify the methods which can be used.

7.1.7 Circle relative

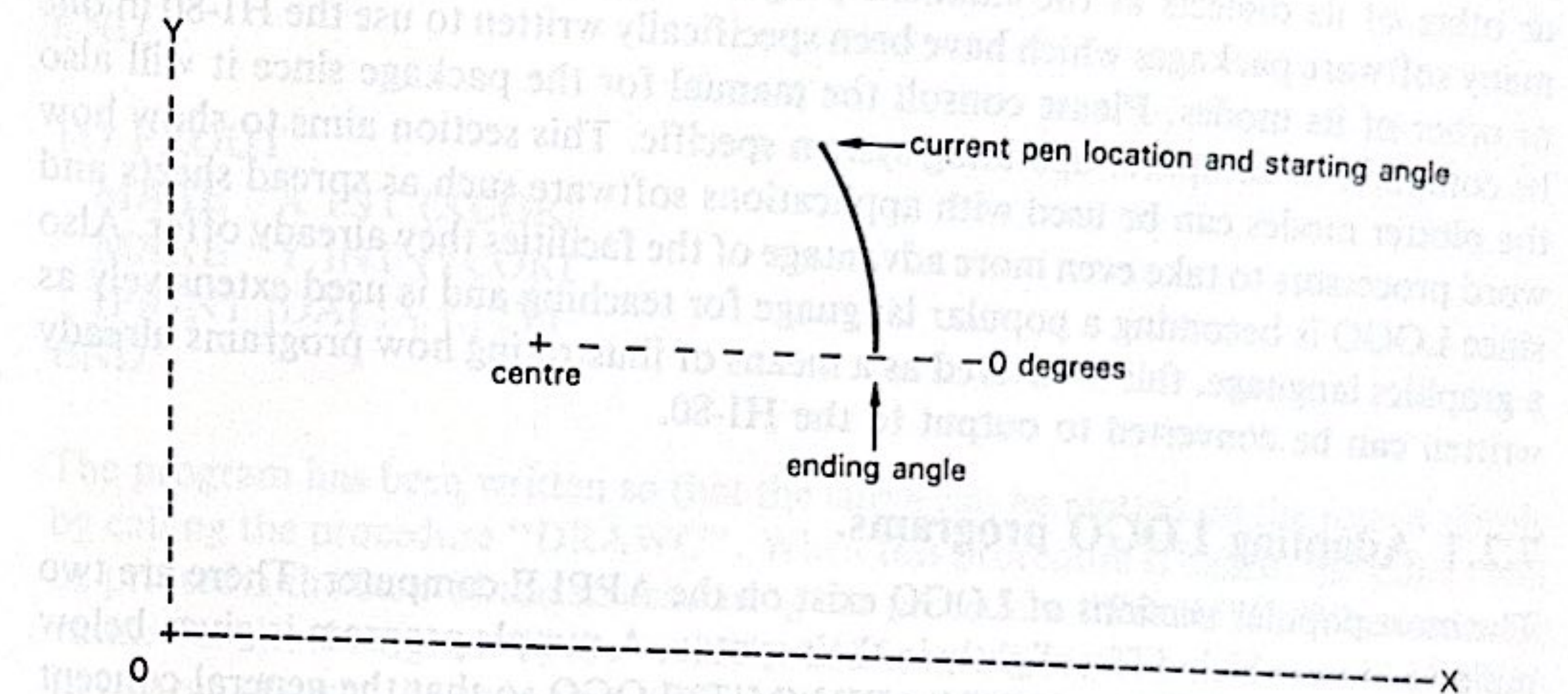
In this case the centre of the circle is not given and the arc is drawn beginning at the current pen location. You must take more care as it is necessary to must work out in which direction you want the arc to go having first decided the position of the centre.



Probably the easiest way to visualise this is to draw 'imaginary' radii from the starting and ending angles to the centre. It is then easy to calculate the angles in relation to the radius line at 0 degrees (parallel to the X-axis and through the supposed centre of the circle). In the example above these 'imaginary' lines are the dotted lines. You can now calculate the angles in the same way as before. For this example, θS will be 1200 (120 degrees anticlockwise from the 0 degrees radius line) and θE will be 1700 (170 degrees anticlockwise from the 0 degrees radius line). The amount of curve the arc has is determined by the radius itself, that is, a large radius will result in a gently curving arc and a small radius will result in a very tightly curving arc. To produce the above example a radius of about 2.5cm ($r = 250$) will be required, resulting in the following command values:

`"CR 250,1200,1700"`

In some circumstances it may be easier to execute a "CA" command to draw the arc because of the difficulty of estimating the starting and finishing angles, for instance when drawing a picture rather than mathematically defined graphics. The absolute case is easier to handle simply because the points are definite positions and do not require any further computation.



Here, instead of executing a "CR" command it is probably easier to estimate the location of the centre of the circle of which the arc forms a part, then execute a "CA" command. For instance, the relevant commands could be:

`"CA 500, 600, 300, 300, -3600"`

or

`"CR 300, 300, -3600"`

both of which will produce the desired result. The -3600 is obtained because it is not possible to give a parameter of -0 . The arc is actually being drawn in a clockwise direction so the ending angle must be given as "minus 360 degrees" instead of "minus zero degrees".

7.1.8 Hatching a pie segment

The same methods as used in the preceding examples can be used when using the "HP" (Hatching a Pie segment) command. In this command the centre of the circle which contains the pie is defined by absolute coordinates, so the angles for starting and ending can be calculated in just the same way as with the "CA" command as explained above.

7.1.9 Character direction

The "DI" command can be used to set the direction of rotation for characters printed by the "LA", "LU", "AM" and "RM" commands. The parameter "n" is calculated in the same way as the parameter θS as explained above for the "CA" command.

7.2 Using the HI-80 with Applications Software

The examples in this manual have generally been centred around the BASIC programming language for the simple reason that most microcomputers use BASIC in one or other of its dialects as the standard programming language. There will also be many software packages which have been specifically written to use the HI-80 in one or other of its modes. Please consult the manual for the package since it will also be computer or computer operating system specific. This section aims to show how the plotter modes can be used with applications software such as spread sheets and word processors to take even more advantage of the facilities they already offer. Also since LOGO is becoming a popular language for teaching and is used extensively as a graphics language, this is covered as a means of illustrating how programs already written can be converted to output to the HI-80.

7.2.1 Adapting LOGO programs.

The most popular versions of LOGO exist on the APPLE computer. There are two main versions which differ slightly in their syntax. A sample program is given below in both APPLE LOGO and TERRAPIN (MIT) LOGO so that the general concept can be adapted to other versions of the language. The approach taken has been to find out the coordinates of the Turtle and use them as plotting coordinates for the HI-80. This results in previously written programs being easily adapted, since it is only necessary to invoke a new procedure to draw the lines on the plotter. The program used is a favourite LOGO program, the recursive "C" curve.

APPLE LOGO VERSION

```
TO PROGRAM :SIZE :LEVEL
  FENCE
  SETUP
  .PRINTER 1
  DRAWC :SIZE :LEVEL
  .PRINTER 0
END
```

```
TO SETUP
  .PRINTER 1
  PR [DF]
  PR [IF 1,1]
  PR [MA 1255 , 960]
  PR [OR]
  PR [IM 31, 11]
  PR [IF 10, 1]
  .PRINTER 0
END
```

```
TO DRAWC :SIZE :LEVEL
  IF :LEVEL = 0 [FD :SIZE PROUT STOP]
  DRAWC :SIZE :LEVEL - 1
  RT 90
  DRAWC :SIZE :LEVEL - 1
  LT 90
END
```

```
TO PROUT
  MAKE "X INT (XCOR)
  MAKE "Y INT (YCOR)
  (PRINT [DA] :X [,] :Y)
END
```

The program has been written so that the curve can be plotted on the screen simply by calling the procedure "DRAWC". When this procedure is called, the points will be printed at the base of the screen, using the procedure "PROUT". When suitable values for the side (given by the variable :SIZE) and the level of recursion (:LEVEL) have been chosen, the procedure "PROGRAM" can be used to plot the Turtle's path on the HI-80.

The procedure "PROUT" simply assigns the coordinates of the Turtle at the end of each movement forward and prints out the correct string to move the pen on the HI-80.

The procedure "PROGRAM" controls both output to the printer and calls the printer setup procedure "SETUP". This procedure opens the printer port, then prints the strings "DF" and "IF 1,1" to set the plotter to the default state, and ensure that the magnification is set to normal. The pen is moved to the centre of the paper since the Turtle is set at the centre of the screen, using the "MA 1255, 960" and "OR" commands. The error handling is set to carry out full trapping and the magnification is set to 10:1 with the "IF 10,1" command.

The procedure "PROGRAM" then opens the printer port to allow the coordinates printed during the execution of the "DRAWC" procedure to be acted upon by the HI-80 and finally switches off the printer port when the curve has been drawn. Also the "FENCE" command is given so that any errors where the curve wraps around the screen are trapped if the curve has not been tested simply on the screen. If this is not done and the curve does wrap around, a number of horizontal lines will be drawn on the plotter as the plotted points move within the coordinates of the computer screen.

The MIT version of the program is as follows:

TERRAPIN LOGO VERSION

```
TO TPROGRAM :SIZE :LEVEL
  NOWRAP
  HOME
  SETUP
  OUTDEV 1
  TDRAWC :SIZE :LEVEL
  OUTDEV 0
END
```

```
TO SETUP
  OUTDEV 1
  PR [DF]
  PR [IF 1,1]
  PR [MA 1255 , 960]
  PR [OR]
  PR [IM 31, 11]
  IF 10, 1]
  OUTDEV 0
END
```

```
TO TDRAWC :SIZE :LEVEL
  IF :LEVEL = 0 FD :SIZE PROUT STOP
  TDRAWC :SIZE :LEVEL - 1
  RT 90
  TDRAWC :SIZE :LEVEL - 1
  LT 90
END
```

```
TO PROUT
  MAKE "X INTEGER (XCOR)
  MAKE "Y INTEGER (YCOR)
  (PRINT [DA] :X [,] :Y)
END
```

Apart from the main syntax differences, the printer port is switched on and off by the "OUTDEV" command rather than the ".PRINTER" command of APPLE LOGO.

7.2.2 Using the HI-80 with a Word processor

The use of the plotter mode with a word processor is simply a matter of typing the commands as the series of commands you would expect any other program to output. For example if the word processor file is as follows:

```
MA 1200, 960
OR
MA 100,100
DR 0, -200
DR -200,0
DR 0, 200
DR 200, 0
```

Simply printing the file as if it was a normal text file will cause the operations to be performed as follows. The plotter will move the pen to location (1200,960), then set the origin there. The next command moves the pen to the top of a square and draws the around the four sides.

To use the word processor in text mode at the same time, the word processor must have the facility to send control characters. This is possible with PEACHTEXT and APPLEWRITER II, but not with WORDSTAR unless it has been specially modified. If you wish to mix text and graphics with any word processor, the best procedure is to print the graphics first, then switch the HI-80 off and switch it back on in text mode by holding the CAP/OFF key down. Then print the text. A word processor capable of calculation, or the WPL (Word processing Language) facility of APPLEWRITER II can be used to program the position of the plotter and give even more versatility.

7.2.3 Using the HI-80 with a Spreadsheet program.

The spreadsheet programs, of which SUPERCALC and VISICALC are the best known, are programs which involve a great deal of calculation and so are ideally suited for use with the plotter modes of the HI-80. Data can be saved as text files and printed out using word processor programs. The spreadsheet can be used directly also, and so an example has been given to explain how to use them. This example has been written in SUPERCALC.

7.3 Some Example Programs

The following programs in MICROSOFT BASIC and using spreadsheets are given as examples of the use of the HI-80. They show some of the capabilities of the plotter, but also some of the techniques for writing programs which differ from plotting on the computer screen. It is useful for BASIC programmers to investigate the SUPERCALC example and vice versa since the techniques used are very similar.

7.3.1 Lines drawing

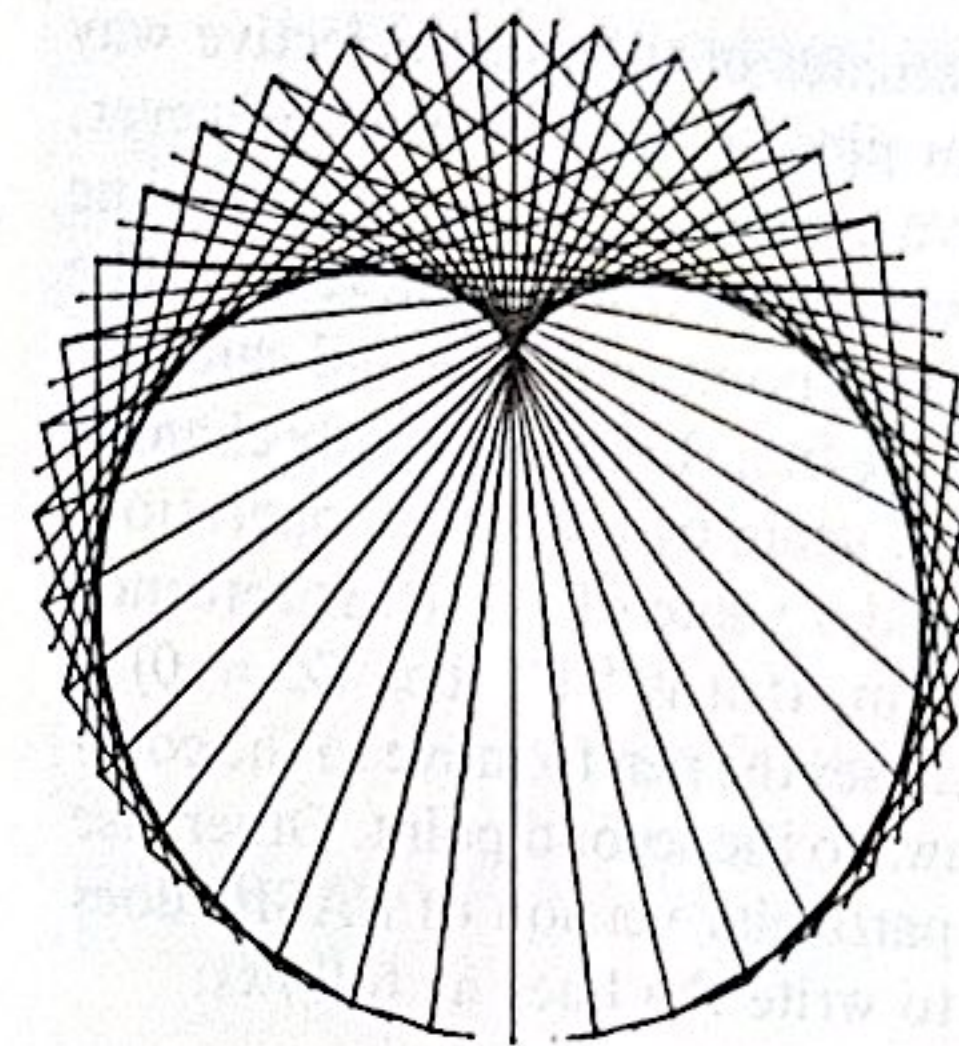
This program illustrates how a curve can be drawn which is made up of a series of tangents. It is described in order to show a number of programming points including some possible errors.

The program calculates a number of points on a circle and joins the points with lines in such a way as one end of the line moves twice as fast as the other. By altering the variable Q in line 50 different curves can be drawn as the diagram illustrates.

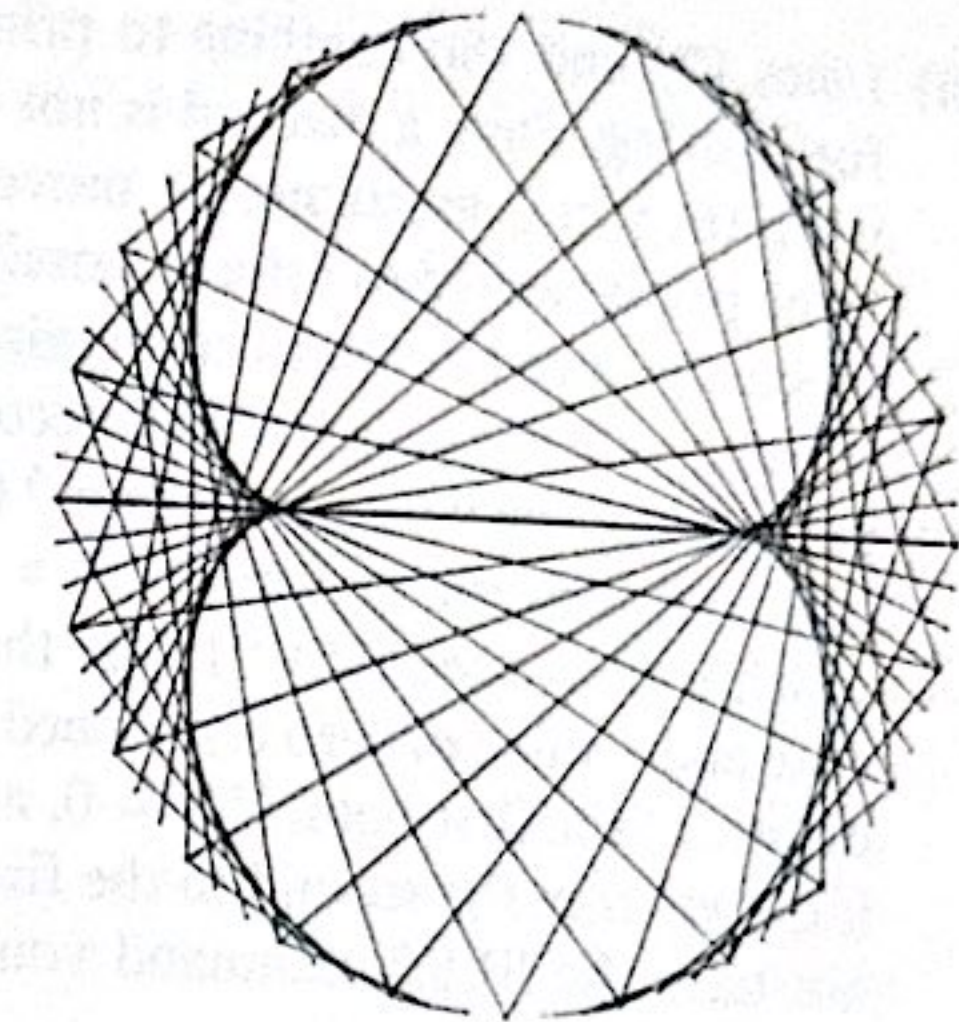
```

5 LPRINT "IF 1,3" - move to line 90 to set picture in middle of
10 'Curves from lines sheet - otherwise parameters are
20 PI = 4 * ATN(1) shrunken by 1/8 by this command
30 R = 900 : 'radius of circle
40 M = 72 : 'number of points on circle
50 Q = 2 : 'parameter which specifies type of curve
60 W = 2 * PI/M : 'defines angle between points on circle
70 '
80 LPRINT "MA 1255, 960" : LPRINT "OR"
90 '
100 FOR J = 1 TO M
110 ANG1 = W * J : 'angle for first point
120 ANG2 = Q * ANG1 : 'angle for second point
130 X1 = INT(R * COS (ANG1)) : 'coordinates of first point
140 Y1 = INT(R * SIN (ANG1)) : 'coordinates of first point
150 X2 = INT(R * SIN (ANG2)) : 'coordinates of second point
160 Y2 = INT(R * SIN (ANG2)) : 'coordinates of second point
170 IF Z = 0 THEN LPRINT "MA"; X1; ","; Y1 : LPRINT "DA"; X2; ","; Y2
    ELSE LPRINT "MA"; X2; ","; Y2 : LPRINT "DA"; X1; ","; Y1
180 Z = - (Z=0)
190 NEXT J
200 LPRINT "DF"

```



Q=2



Q=3

Lines 20 to 60 define a number of variables, which are shown in the REM statements. Line 20 calculates a value for PI the ratio of the circumference to the diameter of a circle. The points of the circle are calculated as polar coordinates and converted to Cartesian coordinates by lines 130 and 140. It is not necessary to understand the mathematics to understand the program. Line 60 calculates the fixed angle between each point on the circle.

Line 80 moves the pen to the centre of the paper and then sets the origin there. This makes it more convenient to calculate the points of the circle.

Lines 100 to 190 form a loop which calculates the coordinates of the ends of the lines and joins them. There are two points to note here which can avoid problems in your programs.

- (i) The coordinates are found by taking the integer of the calculated value. This is because BASIC sometimes calculates very low values and expresses them in scientific notation, for example "3.013246E-04". In such a situation the HI-80 will treat the "E" of the exponential value as an incorrect character and generate an error. Taking the integer overcomes this problem.

- (ii) Lines 170 and 180 combine to print the coordinates in the most effective way for plotting. Such a method is not required for plotting on a screen. However, when the pen is mechanically moved it can save a considerable amount of time if the pen is moved as little as possible before starting to draw the next line. The procedure is to alternate the drawing from the first point to the second calculated point with drawing from the second point to the first. Line 180 simply changes the value of the variable Z from 0 to 1 and back again by a Boolean operation. If $Z = 0$ then the statement $(Z = 0)$ is true and a value of "-1" is returned which is converted into "1" by the negative sign. If it is "1" then $(Z = 0)$ is false and a value of zero is returned. Line 170 causes the pen to move to the coordinates of the first point if $Z = 0$, and then draws to the second point. Otherwise it draws from the second to the first. If your particular version of BASIC does not have the ELSE command you will have to write the lines as follows:

```
170 IF Z = 0 THEN LPRINT "MA"; X1; ","; Y1 : LPRINT "DA"; X2
; ","; Y2 : GOTO 180
```

```
175 LPRINT "MA"; X2; ","; Y2 : LPRINT "DA"; X1; ","; Y1
```

The last line of the program resets all parameters to their default values so that a new program will run correctly.

7.3.2 A Cartoon using curves.

The cartoon for this program was sketched on paper first and the positions of points determined by measurement. The following description is meant to show you why various techniques were adopted.

```
10 : *** CARTOON PROGRAM ***
20 :
30 LPRINT "DF"
40 LPRINT "IM31,3"
50 WIDTH LPRINT 255
60 :
70 *** COORDINATES OF TOP LEFT HAND END OF WALL ***
80 :
90 X = 1350 : Y = 0
100 :
110 *** MIDDLE OF WALL ***
120 :
130 LPRINT "MA1730,1920"
140 LPRINT "DA1730,0"
150 LPRINT "MA1750,0"
160 :
170 *** BOTTOM OF WALL ***
180 :
190 LPRINT "BR135,7,240,8"
200 :
210 *** TOP OF WALL ***
220 :
230 A$ = "MR0,200"
240 B$ = "DR0,350"
250 LPRINT "MA"; X; ","; Y
260 LPRINT "DR0,300"
270 LPRINT A$
280 LPRINT B$
290 LPRINT A$
300 LPRINT B$
310 LPRINT A$
320 LPRINT "DA"; X; ","; Y
330 :
340 *** HEAD ***
350 :
360 LPRINT "VA0,1350,550,1050,550,900,600,800,700,750,800,700,950,750,1100,800,1
200,900,1300,1050,1350,1350,1350"
370 :
380 *** HAIR ***
390 :
400 C$ = "MA800,1200"
410 LPRINT C$
420 LPRINT "CR150,-1350,-2250"
430 LPRINT C$
440 LPRINT "CR200,450,1350"
450 LPRINT C$
460 LPRINT "CR100,450,1350"
470 :
480 *** EYES AND NOSE ***
```



```

490
500 LPRINT "VA0,1350,650,1275,650,1225,665,1190,700,1170,760,1190,800,1225,825,1
275,850"
510 LPRINT "VA0,1275,850,1500,860,1525,870,1550,890,1555,950,1550,1000,1525,1030
,1500,1050"
520 LPRINT "VA0,1500,1050,1275,1050,1210,1075,1180,1100,1170,1140,1190,1200,1225
,1225,1275,1250,1350,1250"

```

```

530
540 *** EYEBROWS ***
550

```

```

560 D$ = "CR125,-1350,-2250"
570 LPRINT "MA1000,680"
580 LPRINT D$
590 LPRINT "MA1000,1080"
600 LPRINT D$

```

```

610
620 *** PUPILS ***
630

```

```

640 LPRINT "HP2," ; X ; ",.750,75,900,2700,1.0"
650 LPRINT "HP2," ; X ; ",.1150,75,900,2700,1.0"

```

```

660
670 *** HANDS ***
680

```

```

690 FOR J = 300 TO 500 STEP 50
700 GOSUB 900
710 NEXT
720 FOR J = 1400 TO 1600 STEP 50
730 GOSUB 900
740 NEXT

```

```

750 E$ = "CR25,-900,-2700"
760 LPRINT "MA" ; X ; ",.300"
770 GOSUB 960
780 LPRINT "MA" ; X ; ",.1400"
790 GOSUB 960

```

```

800 E$ = "CR25,-900,900"
810 LPRINT "MA1425,1400"
820 GOSUB 960
830 LPRINT "MA1425,300"
840 GOSUB 960
850 LPRINT "DF"
860 END

```

```

870
880 *** SUBROUTINE TO DRAW DIVISIONS BETWEEN FINGERS ***
890

```

```

900 LPRINT "MA" ; X ; ",." ; J
910 LPRINT "DR75," ; Y
920 RETURN
930

```

```

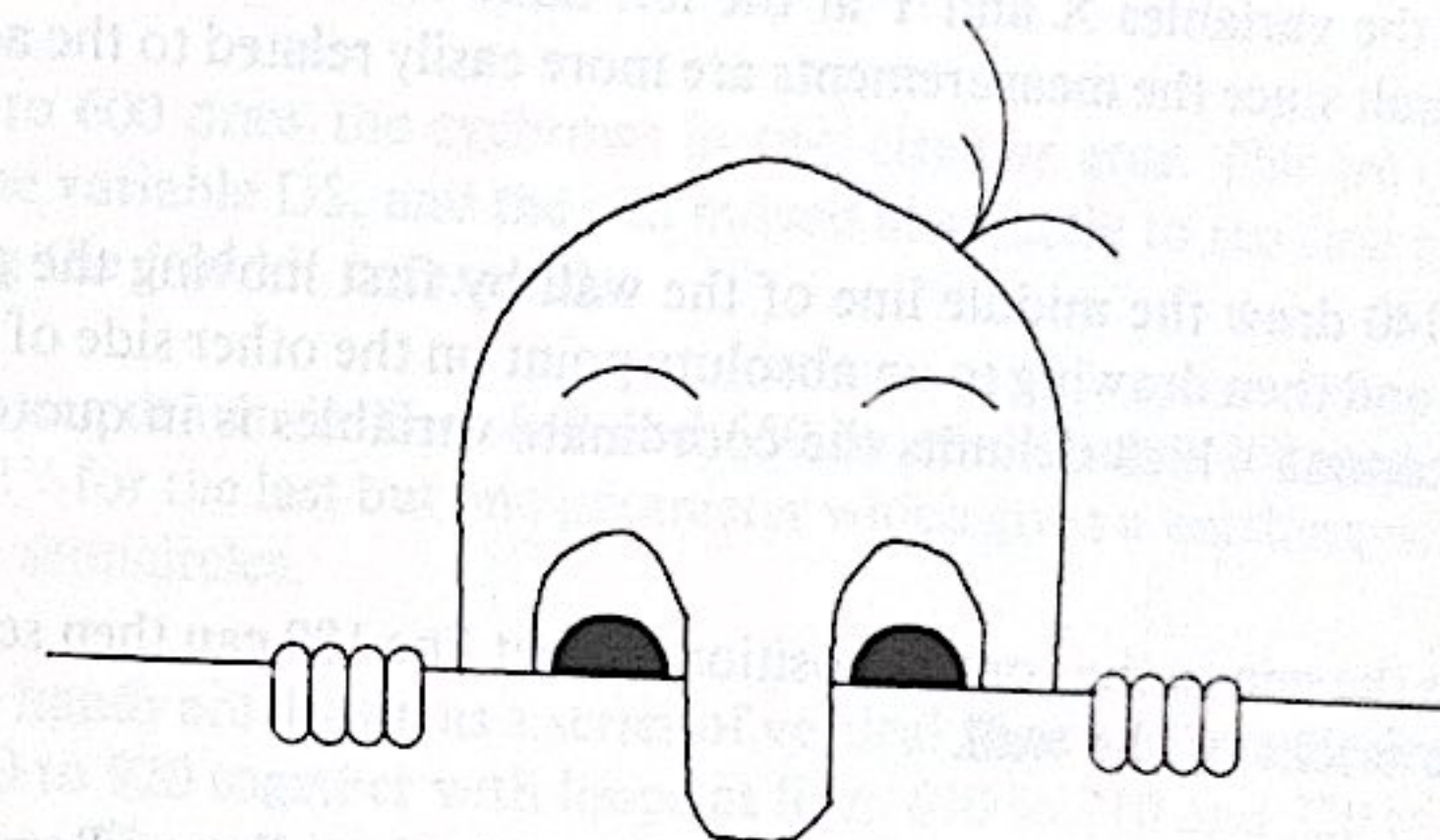
940 *** SUBROUTINE TO DRAW ENDS OF FINGERS AND KNUCKLES ***

```

```

950
960 FOR J = 1 TO 4
970 LPRINT E$
980 NEXT
990 RETURN

```



(Shown one half actual size.)

Lines 30 to 50 set up the HI-80 and the computer for printing. The HI-80 is first set so that all parameters have their default values. Error making is then set so that if there are any errors generated when typing the program in the HI-80 will print them out. Finally in line 50, the MICROSOFT BASIC command is given to allow infinite characters to be sent to the printer without sending an automatic carriage return.

Line 90 sets up the variables X and Y at the left hand side of the wall. The origin is left as the default since the measurements are more easily related to the actual piece of paper.

Lines 130 and 140 draw the middle line of the wall by first moving the pen to one end of the line, and then drawing to an absolute point on the other side of the paper. Note how the comma which delimits the coordinate variables is in quotes between the variables.

Line 150 moves the pen to the correct position so that line 180 can then set up a grid to simulate the bricks of the wall.

In drawing the top of the wall it is necessary to make four lines, allowing breaks for the hands and nose. This is mainly done by a series of relative movements. The strings AS and BS are defined to give relative movements once the pen has been moved to the edge of the wall in line 250. The pen draws the first part of the wall top with a relative command (line 260) and then the pen is successively moved (by printing AS) and then a line is drawn (by printing BS). Finally the last part of the line is drawn to the edge of the effective drawing area using an absolute draw command in line 320.

The next part of the program uses the Curve Absolute command to draw the head. Random points of the head were chosen and their coordinates measured. Where the curve was not quite smooth, another point was added. The parameter "0" immediately after the "VA" of the command signifies that the curves is to be open ended, i.e. the first and last points are not to be joined.

The hair is drawn next using a series of arcs of circles. The point from which the hair grows is defined as a variable CS. The pen is positioned at this point before drawing each arc, by printing CS. Each circular arc is drawn using the "CR" command since this relative command draws the arc from the current pen position. The first arc in line 410 is drawn in a clockwise direction by using negative values of the angles. The other arcs are drawn using an anticlockwise drawing movement.

The eyes and nose are drawn using a continuous curve using three command sequences in lines 500 to 520. Although line 50 has carried out the command to prevent BASIC adding carriage returns and line feeds, it was found that the operating system added them when the computer ran the program. This necessitated the same point having to be specified at the end of line 500 and the beginning of line 510, with coordinates(1275,850). A similar course of action has to be taken in at the end of line 510 and the beginning of line 520 with point (1500,1050). If this is not done, the curve would have breaks in it.

Lines 560 to 600 draw the eyebrows as two circular arcs. The arc command is defined by the variable DS, and the pen moved absolutely to the first point of the eyebrow using the "MA" command.

The pupils are drawn in lines 640 and 650 by specifying a hatched pie but using a value of "1" for the last but one parameter which gives a hatching which completely fills in the semicircles.

Finally the hands are drawn as a series of vertical lines and semicircles. A subroutine at lines 900 to 920 together with loops at lines 690 to 710 and 720 to 740 draws the vertical lines. The pen is moved to the absolute position of the tops of the line and a line is drawn relatively downwards. The circles for the knuckles and the ends of the fingers are defined as the variable ES. Since the "CR" command draws the arc from the current pen position, it is only necessary to position the pen at the end of the series of arcs each time and then use the loop in subroutine 960 to 990 to draw four successive arcs.

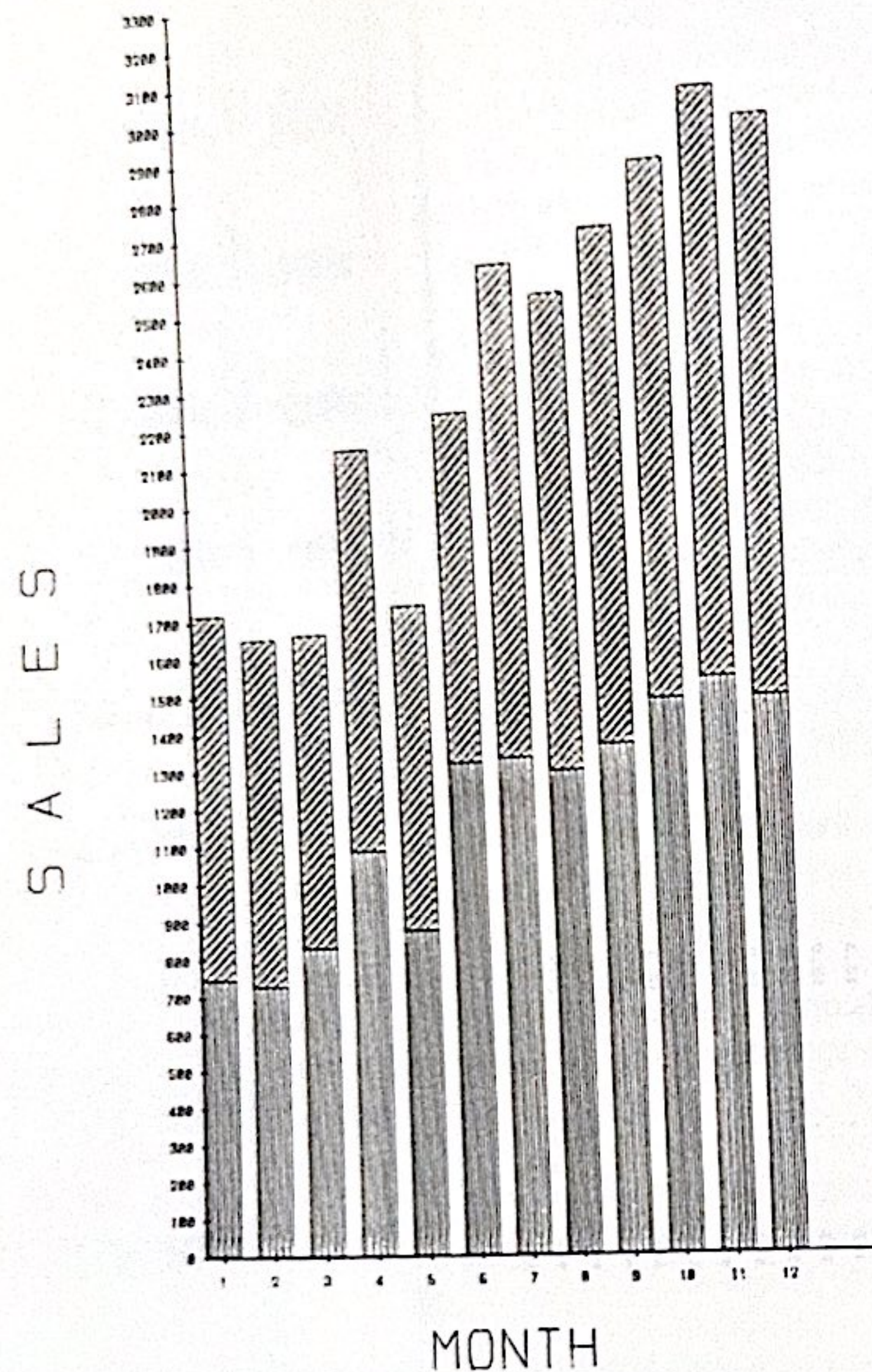
The final command is to set the parameters back to the default. It is good practice to do this at the end of every program as it can prevent problems such as the pen being at an incorrect position for the next sheet of paper and the magnification factor being wrong.

7.3.3 A Graph plotted from SUPERCALC

The following example shows how to use the data from a SUPERCALC spreadsheet to obtain graphical output from the data. If you are using another spreadsheet such as VISICALC similar rules will apply in setting it up, but the commands may be slightly different. The example uses a simple set of sales figures obtained by two sales areas over an annual period. The totals for each month for the two areas are plotted as a bar chart with appropriate labelling. On the screen the spreadsheet looks as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	SALESMAN													
2														
3														
4	AREA A													
5	111	165	125	190	185	190	233	199	255	245	267	289	309	
6	123	187	199	176	176	176	233	234	234	290	299	298	312	
7	222	129	204	210	210	167	245	289	289	290	303	290	290	
8	189	115	177	219	219	176	278	277	278	301	298	304	294	
9	69	115	120	222	222	155	209	299	219	207	278	304	245	
10														
11	TOTAL	SUM(C7:C11)	SUM(D7:D11)	SUM(E7:E11)	SUM(F7:F11)	SUM(G7:G11)	SUM(H7:H11)	SUM(I7:I11)	SUM(J7:J11)	SUM(K7:K11)	SUM(L7:L11)	SUM(M7:M11)	SUM(N7:N11)	
12														
13	AREA B													
14	101	190	135	189	189	188	189	206	257	278	273	306	289	
15	190	176	157	176	176	134	157	278	281	278	219	302	297	
16	289	206	156	123	123	145	168	265	278	292	289	293	289	
17	156	167	176	289	289	201	167	254	201	256	305	312	289	
18	234	183	205	289	289	201	189	271	211	290	305	312	289	
19														
20	TOTAL	SUM(C16:C20)	SUM(D16:D20)	SUM(E16:E20)	SUM(F16:F20)	SUM(G16:G20)	SUM(H16:H20)	SUM(I16:I20)	SUM(J16:J20)	SUM(K16:K20)	SUM(L16:L20)	SUM(M16:M20)	SUM(N16:N20)	
21														
22														
23														
24														
25	HI-80													
26	2	150												
27	100													
28														
29	CI													
30	IND1.11													
31	IF1.2													
32	SP1													
33														
34	MA	700	400											
35	A2.3300.33.0.0.3300.30.1													
36	MA	B34	D34											
37	DR	B27+13	D36											
38	MA	B36+(B28/2)	11.0.1.1.50.0											
39	MA	B27												
40	100													
41	MA	B34+(B37/3)	D34-300											
42	LAONTH		D34+1000											
43	MA	B34-400												
44	DI900													
45	LA S A L E S		D34											
46	MA	B34												
47	DI 0		2500											
48	LAAREA B		1500											
49	MA	B48												
50	MA													
51	LAAREA A													
52	SP2													
53	MA	B34	D34											
54	MA	B27	D34+FS4											
55	MA	B27	D34											
56	MA	B27	D34											
57	MA	B27	D34											
58	MA	B27	D34											
59	MA	B27	D34											
60	MA	B27	D34											
61	MA	B27	D34											
62	MA	B27	D34											
63	MA	B27	D34											
64	MA	B27	D34											
65	MA	B27	D34											
66	MA	B27	D34											
67	MA	B27	D34											
68	MA	B27	D34											
69	MA	B27	D34											
70	MA	B27	D34											
71	MA	B27	D34											
72	MA	B27	D34											
73	MA	B27	D34											
74	MA	B27	D34											
75	MA	B27	D34											
76	MA	B27	D34											
77	MA	B27	D34											
78	MA	B27	D34											
79	MA	B27	D34											
80	MA	B27	D34											
81	MA	B27	D34											
82	MA	B27	D34											
83	MA	B27	D34											
84	MA	B27	D34											
85	MA	B27	D34											
86	MA	B27	D34											
87	MA	B27	D34											
88	MA	B27	D34											
89	MA	B27	D34											
90	MA	B27	D34											
91	MA	B27	D34											
92	MA	B27	D34											
93	MA	B27	D34											
94	MA	B27	D34											
95	MA	B27	D34											
96	MA	B27	D34											
97	MA	B27	D34											
98	MA	B27	D34											
99	MA	B27	D34											
100	MA	B27	D34											
101	MA	B27	D34											

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
59	MA	B57		D57+FS8		D22								
60	MA	A27		B28		E13								
61	MA	B59+B27		D34		E22								
62	MA	B60+B32		D34		F13								
63	MA	A27		D34		F22								
64	MA	B63+B27		D34		G13								
65	MA	D65		D34		G22								
66	MA	B67+B27		D34		H13								
67	MA	A27		D34		H22								
68	MA	B69		D34		I13								
69	MA	B71		D34		I22								
70	MA	A27		D34		J13								
71	MA	B73		D34		J22								
72	MA	B75		D34		K13								
73	MA	A27		D34		K22								
74	MA	B77		D34		L13								
75	MA	A27		D34		L22								
76	MA	B79		D34		M13								
77	MA	A27		D34		M22								
78	MA	B81		D34		N13								
79	MA	A27		D34		N22								
80	MA	B83		D34										
81	MA	A27		D34										
82	MA	B85		D34										
83	MA	A27		D34										
84	MA	B87		D34										
85	MA	A27		D34										
86	MA	B89		D34										
87	MA	A27		D34										
88	MA	B91		D34										
89	MA	A27		D34										
90	MA	B93		D34										
91	MA	A27		D34										
92	MA	B95		D34										
93	MA	A27		D34										
94	MA	B97		D34										
95	MA	A27		D34										
96	MA	B99		D34										
97	MA	A27		D34										
98	MA	B101		D34										
99	MA	A27		D34										
100	MA	B103		D34										
101	MA	A27		D34										



Lines up to 24 are the data, with the totals to be used for plotting in rows 13 and 22.

Setting up the rest of the data for plotting is relatively easy but like many descriptions of how to enter data may look involved. When you come to try out the following example it will probably take you longer to enter the data for the sales figures than it will to set up the plotting data.

Rows 27 and 28 contain constants which can be altered to allow flexibility in producing various graphs. The box A27 contains a value which decides the type of bar to be drawn by the "HB" command. It is set to the value "2" in the example; this will produce a box and hatch it. Box B27 contains the value for the separation of the bars and box B28 the width of the bars.

The commands to be sent to the plotter begin at row 29.

Rows 29 to 32 configure the plotter. The interface is cleared of any errors (using the "CI" command) and then an error handling routine is set up (with "IM37,11"). The magnification is reduced to one half using the "IF1,2" command; this also changes the size of the text characters. The pen to draw the axes is selected using the "SP1" command.

Row 34 selects the origin for the axes. The method of specifying this is to first to place the "MA" command in box A34, then the X coordinate as a value in box B34, then using text a "," in box C34 and finally the value for the Y coordinate in box D34.

Row 35 has the complete command for the Y axis inserted in box A35. It may be more convenient to set up a series of boxes along the row as with the above line.

Row 36 is the first one in which the SUPERCALC commands are used. The row sets the pen position back to the axes origin using the "MA" command. Box A36 simply contains "MA" as a text string. Box B36 then duplicates the contents of B34.

Box B35 has the comma delimiter as a text command and finally box D36 duplicates box D34. Thus if the origin of the axes are altered it is only necessary to change the contents of boxes B34 and D34.

The X axis of the graph is drawn with row 39. However, because the marking of the axis is required equally spaced for the months, the axis is drawn to the right of the origin of the axes as seen in the final result. Row 37 draws a line which is longer than the axis drawn by row 39, so that a better result is obtained. This is carried out by entering the formula $B27*13$ onto box B37 so that row 37 draws a relative line along the position of the X axis of the graph which is 13 times as long as the separation of the bars stored in box B27.

Row 38 moves the pen back along the X axis but to a position which is found as follows. The contents of box B38, which constitute the X coordinate for the "MA" command of row 38, is given by the formula $B36 + (B28/2)$. This means a position half a bar width (the contents of B28 are the bar width) from the origin. The Y coordinate is found by copying the contents of D36 into D36.

Row 39 then draws the X axis of the graph producing the marks for the months. Box A39 is the text "AX1," and then box B39 copies B27 to duplicate the bar width. Boxes C39 and D39 then contain the strings "," and "11,0,1,1,50,0" which will draw the axis as shown.

Row 40 sets the size of characters for the labels on the axes. Box A40 contains the characters "SI 100,100"

Row 41 moves the pen to a position approximately one third of the length of the axis to print the label "MONTH" using the "LAMONTH" command in box A42. The contents of box B41 is given by the formula $B34 + (B37/3)$. The label is moved away from the axis by using the formula $D34 - 300$ in box D41.

Row 43 moves the pen in a similar way to position the pen to print the other label on the Y axis. Box B43 contains the formula $B34 - 400$ to move the label away from the axis. Box D43 contains the formula $D34 + 1000$ to position the start of the label.

Box A44 contains the command "DI 900" to rotate the characters so that they are printed perpendicular to the axes. The label is then printed by the command in box A45.

Duplicate row 34 onto row 46 to return to the origin.

Box A47 returns the characters to their normal position with the "DI 0" command.

Rows 48 and 50 move the pen to print the labels in boxes A49 and A51 respectively. Box B48 contains the formula $B34 + B37 + 300$, D48 the value 2500; box B50 duplicates B48 and D50 contains the value 1500.

Box A52 changes the pen colour.

Row 53 duplicates row 34 to return to the origin.

You can now set up the hatching conditions and present the data for plotting. This is carried out using the following set of commands and the extensive use of replication.

Insert the text "HB" into box A54. Now make the column from A53 to A100 as follows:

Replicate from A53 to A54 to box A55.

Replicate from A53 to A56 to box A57.

Replicate from A53 to A60 to box A61

Replicate from A53 to A68 to box A69

Replicate from A53 to A68 to box A85.

Now transfer the contents of the boxes as follows:

Make B54 the contents of box A27; this sets the bar type

Make B55 the contents of box B34; this sets the X coordinate of the base

Make B56 the contents of box A27; same bar type

Make B57 the contents of box B55 + B27; offset by the bar width

Make B58 the contents of box A27; same bar type

Make B59 the contents of box B57; the same X coordinate

Now make the column from B60 to B100 as follows but with adjustments on all values except where they are the constants in boxes A27 and B27.

Replicate from B56 to B59 to box B60

Replicate from B56 to B63 to box B64

Replicate from B56 to B71 to box B72

Replicate from B56 to B68 to box B88

Now make the columns of commas. Note that column C has commas in every row, but columns E and G have them only on the rows containing "HB" commands. This is a standard replication procedure.

Row 54 can now be set up to draw the first box.

Make D54 equal to the contents of B28.

Make F54 equal to C13

Put the string "10,0" into box H54 and at the same time "10,3" into box H56. These specify the pitch of the hatching and the direction of hatching. If you prefer change it to be a constant by using a box like A27 which holds the bar type. The contents of H54 will define the hatching for Area A of the sales force, and H56 for Area B.

Row 55 moves the pen to draw the box on top of the box for sales area A. Box B55 must be made equal to the contents of box B53 to give the same X coordinate and then box D55 is filled with the formula $D53 + F54$. This adds the value of sales area A so that the bar is plotted on top.

Box D56 can now be made equal to the value of B28 to set the width of the bars.

Now the rest of column D can be formed by replication as follows using adjustment of all values except where the contents are D34 or B28.

Replicate from D53 to D56 to box D57

Replicate from D53 to D60 to box D61

Replicate from D53 to D68 to box D69

Replicate from D53 to D68 to box D85.

Now column H can be replicated with no need for adjustment.

Replicate from H53 to H56 to box H57

Replicate from H53 to H60 to box H61

Replicate from H53 to H68 to box H69

Replicate from H53 to H68 to box H85.

Column F is slightly harder to replicate. It requires the values to be taken from rows 13 and 22. However, it is logically done so is almost as easy to do as a replication.

Add row 101 to contain "DF" as text in box A101, and the sheet is complete.

To run the file for printing. First turn off the border otherwise an error will result, then use the output command to the printer, to print boxes A29 to H102. Each time a new line is printed the command is terminated.

7-26

8.1 Cleaning

It is recommended that the HI-80 be cleaned periodically to ensure long operating life.

- **Printer case and cover**

Apply neutral detergent to the printer case and cover, and wipe them with a soft cloth. Do not use alcohol, thinner, or other solvents. Using of such solvents will alter the color of the printer case or cover and erase characters on the printer case; wiping the printer with a hard cloth will damage the surface of the printer case or cover. Do not expose electronic parts or printer mechanism to moisture.

- **Pen caps**

The pen caps can be cleaned using a piece of tissue or cotton wool moistured with alcohol.

- **Platen**

The platen may also be cleaned in the same way if traces of ink contaminate it.

- **Inside the printer**

Eliminate any dust or foreign materials on the printer mechanism and circuit board with a soft brush. Be sure not to damage the electronic parts or any portion of circuit.

8.2 Troubleshooting

Should any problem occur, contact your EPSON dealer as soon as possible.